

AVL Trees

Adelson-Velsky and Landis Trees

- Self-balancing binary search tree (alternative to RBTs)
- Uses rotations to maintain balance

Goal:

- Keep H (height) of tree with N nodes growing in $O(\log N)$

AVL Tree Insertion

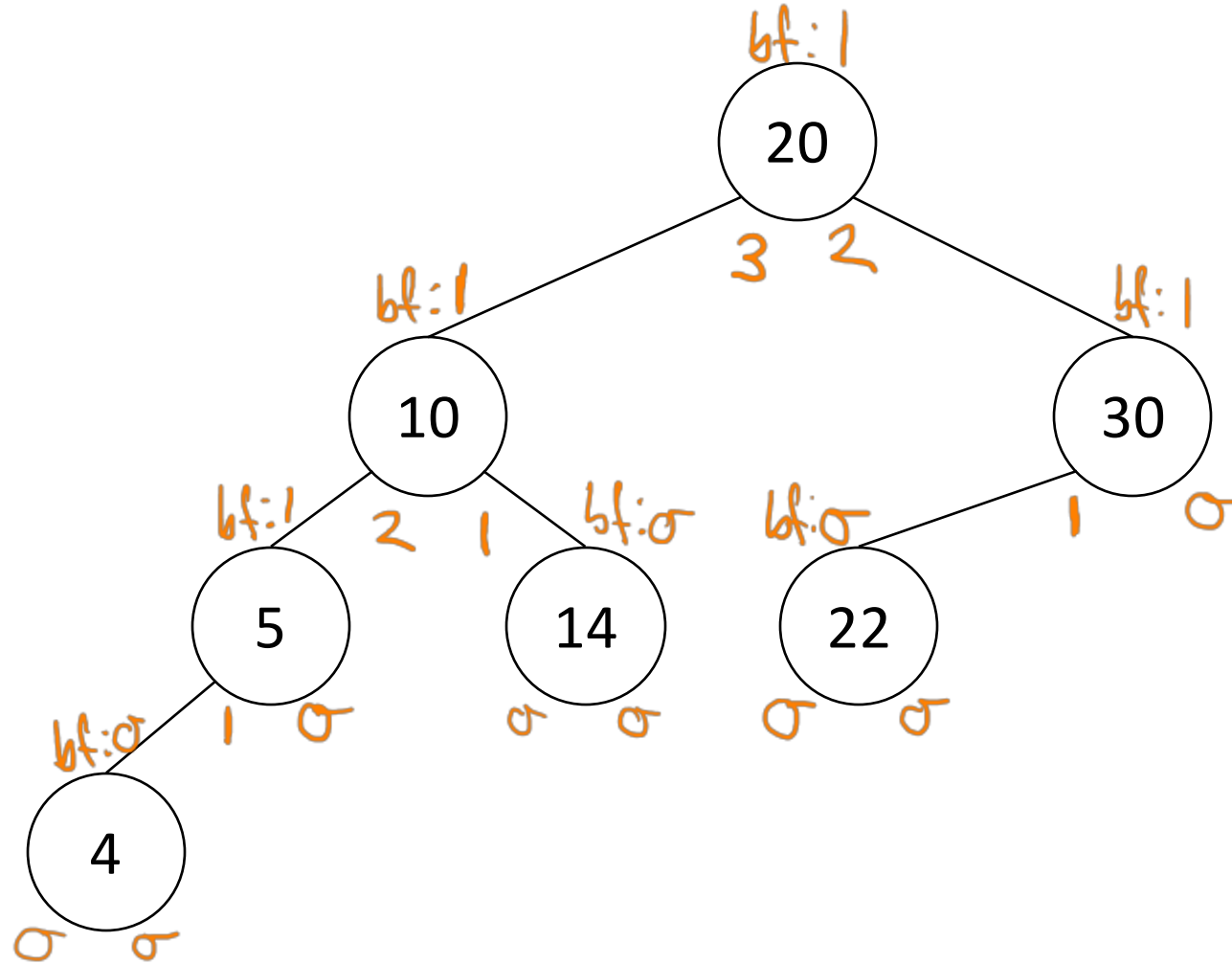
1. Insert new value using regular BST insertion algorithm
2. Check if tree is out of balance, and rebalance if it is

Balance Factor

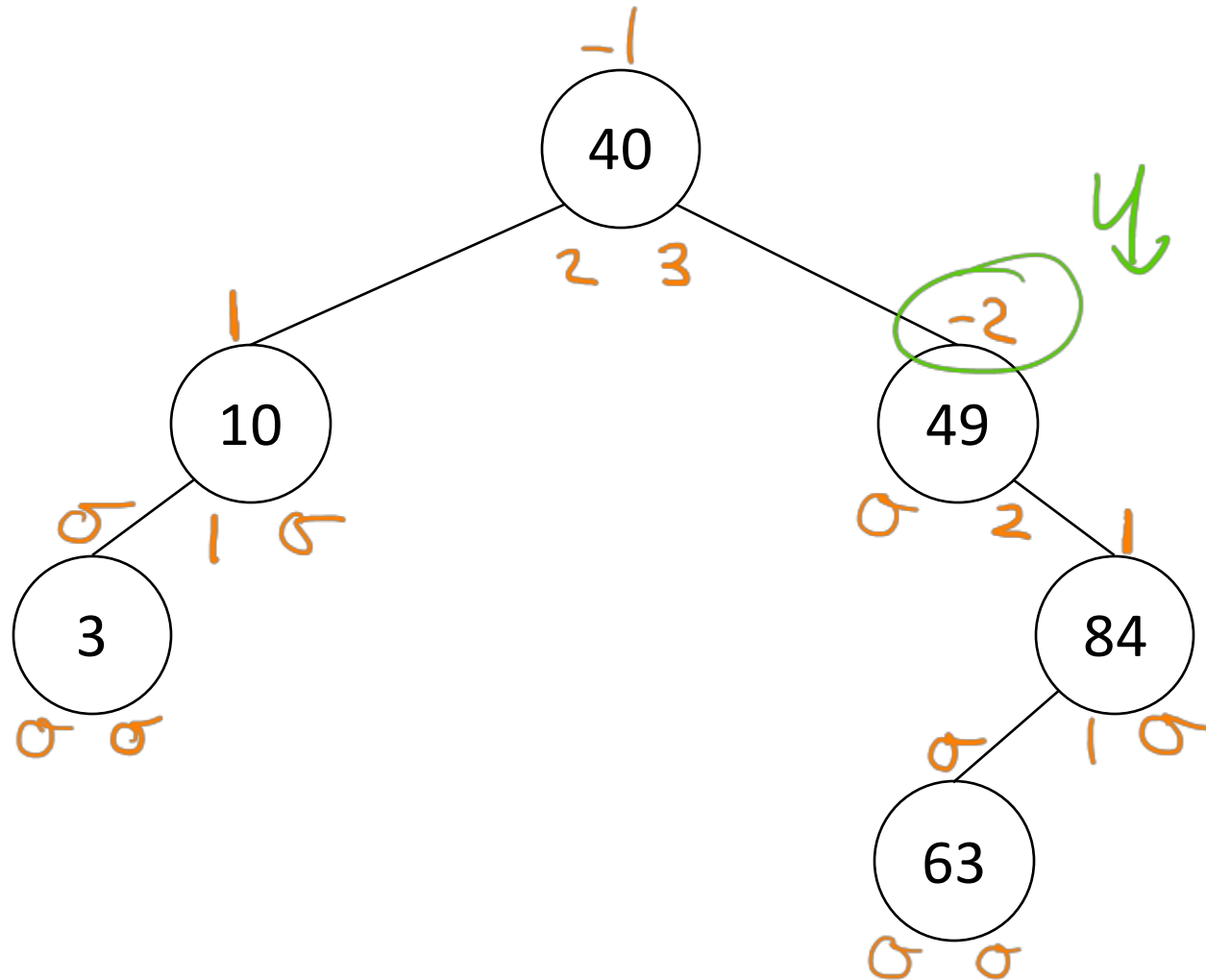
$$bf = \text{height}(\text{left subtree}) - \text{height}(\text{right subtree})$$

- A node is *balanced* if $|bf| < 2$
- Sign tells us direction of imbalance
- A tree is balanced if every node in the tree is balanced

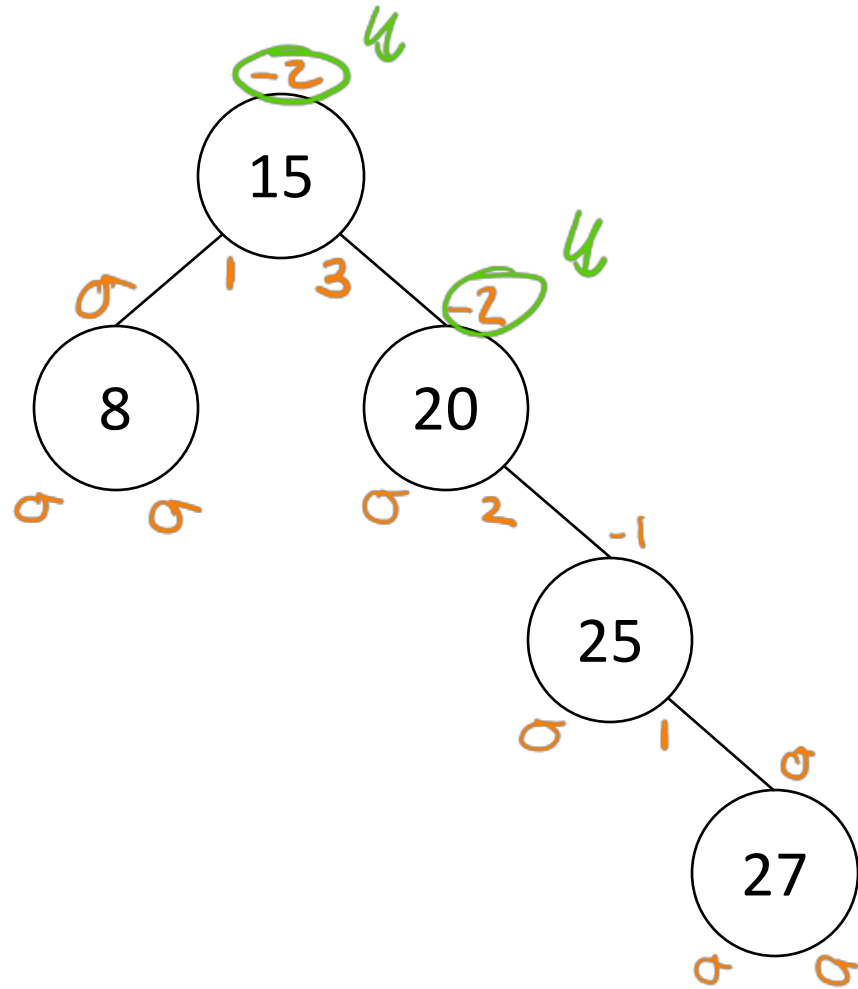
Balance Factor



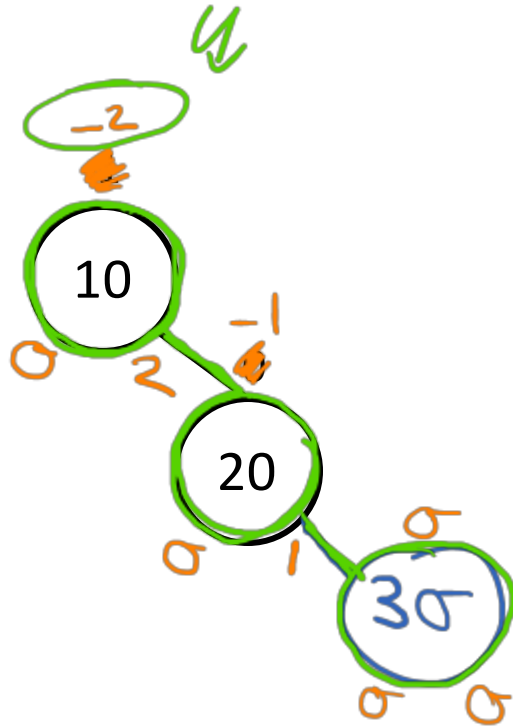
Balance Factor



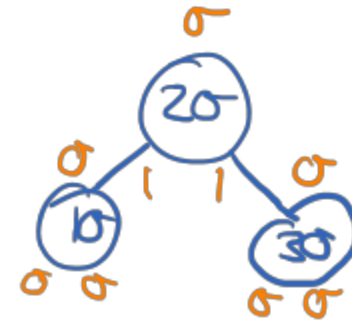
Balance Factor



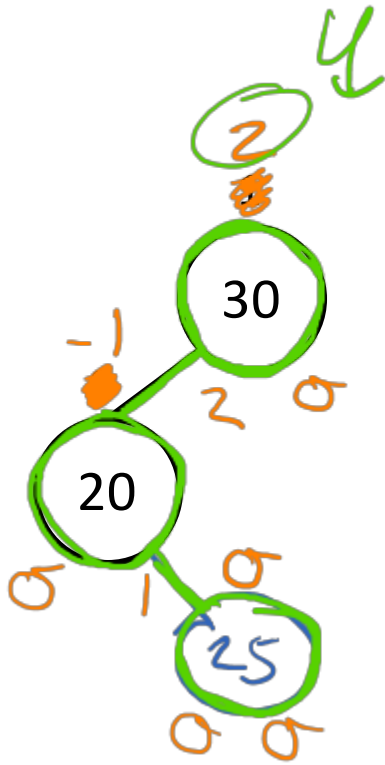
Re-Balancing



left + rot
 \Rightarrow
10, 20

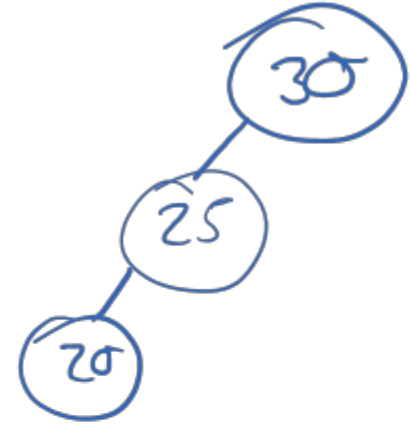


Re-Balancing

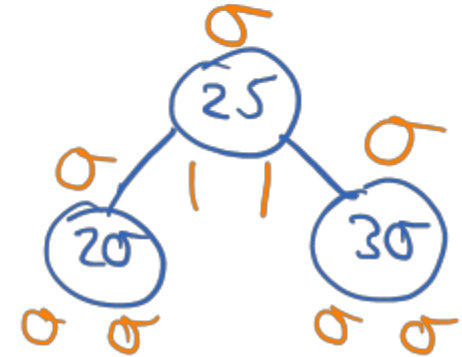


insert: 25

1) left rot
 \Rightarrow
20, 25



2) right rot
 \Rightarrow
30, 25



Re-Balancing

insert: 8

